06

Un unico servlet

Transcripción

[00:00] Hola a todos y todas, bienvenidos y bienvenidas una vez más a este curso de servlet. En este caso les estoy mostrando una diapositiva que había hecho en el curso anterior, en el cual estamos mostrando nuestra división de archivos, nuestra arquitectura dentro de nuestro proyecto.

[00:14] Por un lado, nosotros tenemos lo que son nuestros servlets y modelos, nosotros lo estamos teniendo ahora en un solo paquete, y tenemos la view o las vistas, los jsp por el otro lado, donde sí ellos los tenemos bien separados. Entonces lo que me gustaría hacer ahora en este video es separar servlet de modelo.

[00:36] Entonces vamos a hacer eso. En este caso el tengo como ven, acá está todo junto, está todo mezclado y vamos a separarlo dentro de nuestra DB. Acá arriba vean que dice que el paquete en el que está es gerenciador Servlet, entonces una forma de mover esto a un nuevo paquete es cambiando acá en vez de poner gerenciador.servlet, colocar gerenciador.modelo. ¿Qué es lo que va a ocurrir?

[01:01] Con "Ctrl + S" nos está diciendo Eclipse que no existe ese modelo. Entonces la forma de llevarlo a crear un paquete modelo y mover este archivo DB es con el clic en la lupita y colocar directamente mover DB al paquete modelo. Vean que fue creado el paquete modelo y bien, está dando algunos errores, pero es porque no hemos movido nuestra empresa también para ese paquete que es parte de ese paquete.

[01:30] Entonces vamos a hacerlo fácilmente. Arrastramos empresa hacia el paquete modelo. A ver, no me está dejando, dejen que entonces lo voy a hacer de la otra forma. Vamos a colocar modelo, vamos a hacer el mismo método con la lupita, mover, y ahora ya tenemos nuestra empresa dentro del paquete modelo. ¿Ahora qué es lo que ocurre?

[01:56] Vamos a ir a nuestro Servlet. Y déjenme ver, nos está dando errores acá, nos está diciendo que no está encontrando la base de datos. ¿Por qué? Porque para ese archivo eliminar EmpresaServlet todavía la base de datos está dentro del paquete Servlet, pero nosotros lo movimos el paquete modelo, entonces vamos a solucionar este problema muy fácilmente con un “Ctrl + Shift + O”.

[02:18] Hacemos el import de los archivos y con un "Ctrl + S" guardamos nuestro archivo. Vean que se fueron los errores, vamos a hacer lo mismo con todos los otros. Acá nuestro HolaMundo no usa nuestra base de datos. “Ctrl + Shift + O”, "Ctrl + S", acá también, “Ctrl + Shift + O”, "Ctrl + S" acá y en nuevaEmpresa. Perfecto. “Ctrl + Shift + W” para cerrar todo. Bien.

[02:49] Hemos hecho la reestructuración, hemos movido los archivos que necesitamos para modelo y ahora vean lo siguiente. Hemos hecho esa separación entre modelo y servlet. Y en esta diapositiva ven más o menos cómo es que funciona nuestra aplicación. Estamos viendo que nosotros estamos haciendo una llamada para cada servlet.

[03:06] Una llamada para MostrarEmpresa, una para modificar una, para listar, etcétera. ¿Y qué es lo que ocurre? Lo que a mí me gustaría tener es en realidad una única puerta de entrada a nuestro sistema. Acá es como que cada llamada a un nuevo servlet es una nueva puerta de entrada diferente hacia nuestro sistema. Aquí tenemos varios lugares para ver cada vez que nosotros recibimos una petición.

[03:30] Entonces lo que me gustaría a mí sería tener una única puerta de entrada, un único servlet, que ese servlet sea el encargado de decir: “¿cuáles

acción tengo que realizar dependiendo de cuál es la llamada que yo recibo, cuál es la request que yo recibo?" Eso es lo que me gustaría hacer a mí. ¿Por qué me gustaría hacer esto?

[03:54] Hay varios motivos. Uno de ellos es por seguridad. Haciendo una analogía con nuestras casas, por ejemplo, en nuestras casas, ¿cuántas puertas normalmente tenemos? ¿Una o dos como máximo? Generalmente eso es lo que se tiene. ¿Por qué? Porque nosotros podemos colocar esfuerzos de seguridad en esos puntos de acceso, en las puertas, colocamos un llaves, colocamos trabas, etcétera.

[04:20] Entonces es bueno tener un único lugar donde centralizar nuestros esfuerzos de seguridad y no tener que colocar toda esa seguridad en cada una de las puertas existentes. Entonces, vamos a crear ese único Servlet y el otro motivo también es por el siguiente. Vean por ejemplo nuestro mostrarEmpresaServlet. ¿Nosotros acá para crear ese Servlet qué hacemos?

[04:42] Colocamos un nombre, colocamos un extends httpServlet, leemos parámetros, creamos la base de datos, colocamos un atributo en este caso, llamamos un requestDispatcher, vean nuevaEmpresaServlet. De nuevo extends httpServer tenemos un doPost también, tenemos leemos parámetros.

[04:59] Vean que repetimos acciones, tenemos un patrón dentro de nuestro propio sistema y ese patrón nosotros lo podríamos generalizar y colocar en un único lugar en un único servlet, y luego las peculiaridades de cada acción las colocamos en otro lugar, pero sería bueno tener eso como todo junto dentro de un mismo servler. Entonces, con eso nosotros evitamos lo que se llama boiler plate.

[05:37] Boiler plate es repetir código. Vean que acá tenemos repetición de extends httpServlet, con doPost, tenemos acá también http, doPost, leemos parámetros, entonces, vamos a intentar evitar eso. Y hablando sobre boiler

plate, nosotros dentro de nuestro web.xml por ejemplo, vean que para cada servlet nosotros tenemos muchas líneas de código.

[05:57] Por ejemplo acá en este hola mundo, nosotros tenemos varias líneas de código, en este nuevaEmpresaServlet tenemos también varias líneas de código y para cada uno de ellos vean que tenemos acá 65 líneas de código. No es bueno tener tantas líneas de código repetidas cuando nosotros podemos, por ejemplo, colocar un único servlet que tenga esas líneas de código.

[06:17] Hoy en día es mucho más fácil porque Eclipse ya automáticamente nos está creando este XML. Bien, pero en el pasado era más difícil, ya que teníamos que crear cada una de esas líneas manualmente. Entonces, vean que hay varios beneficios de centralizar los esfuerzos en un único punto, entonces vamos a hacer eso en el próximo video. Nos vemos allá.